AFRL-IF-RS-TR-2004-189
Final Technical Report
July 2004

HUMAN CENTERED, VARIABLE INITIATIVE CONTROL OF COMPLEX AUTOMATA-TEAMS

Cornell University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. M415

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-189 has been reviewed and is approved for publication

APPROVED: /s/

CARL A. DEFRANCO, JR
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** JULY 2004	**3. REPORT TYPE AND DATES COVERED** Final Oct 01 – Dec 03

4. TITLE AND SUBTITLE HUMAN CENTERED, VARIABLE INITIATIVE CONTROL OF COMPLEX AUTOMATA-TEAMS	**5. FUNDING NUMBERS** C - F30602-01-2-0577 PE - 63760E PR - M415 TA - 00 WU - 01
6. AUTHOR(S) Mark Campbell	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Cornell University 208 Upson Hall Ithaca New York 14853	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency, 3701 North Fairfax Drive, Arlington Virginia 22203-1714 AFRL/IFSA, 525 Brooks Road, Rome New York 13441-4505	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER** AFRL-IF-RS-TR-2004-189

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Carl A. DeFranco, Jr./IFSA/(315) 330-3096/ Carl.DeFranco@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT	**12b. DISTRIBUTION CODE**
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT *(Maximum 200 Words)*

A program summary for the Cornell led team in the DARPA MICA program is presented. The solution includes a hierarchical control approach to semi-autonomous systems, research at several levels, including estimation and path planning, strategic and resource planning, and operator interface and decision making. The approach is unique in its use of formal techniques at the lower levels of the hierarchy, optimized/randomized techniques at the higher levels, each with appropriate information fusion and requirements up the hierarchy. Accomplishments are shown in the areas of Streamline Path Planning/Extensions, Cooperative reconnaissance (ISR, RoboFlag system: adoption within community for basic, semi-autonomous research, RoboFlag HitL Studies; initial modeling results, Architecture for Evolution of Pre-planned Strategies and Resource Deployment using GP, Team Tasking using tiered optimization.

14. SUBJECT TERMS Semi-Automated Systems, Command & Control, Decision Making, Planning	**15. NUMBER OF PAGES** 47
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: SUMMARY

The objective of this program was to develop, both theoretically and experimentally, an architecture for realistic semi-autonomous systems composed of human operators and different mobile vehicles using a variable initiative goal setting that achieves a core MICA objective: N vehicles >> M operators. There were four primary elements of this proposal: Human Interface: Developing the human interface requirements, and scaling insights, for M=1 and M>1 operators using cognitive engineering, Variable Initiative Control: Developing variable initiative control strategies with humans in the loop (M=1 and M>1), including novel concepts such as operator "shape" control of the team and football "Playbook" analogies. These are supported by technologies from leveraged programs and those developed here, such as a "World State" ("centralized knowledge, decentralized execution.") concept in an uncertain environment, Experimental Validation using RoboFlag: evaluating algorithms and concepts in an experimental competition called RoboFlag, and Technology Transition: Transitioning the technology to MICA partners as well as the outside community, especially government users such as the US Air Force.

The Cornell team, led by Cornell University and with partners Caltech, Catholic University, and SIFT, worked diligently through the initial milestones in an effort to produce tangible results as quickly as possible. During the 2+ years of the Cornell led program, accomplishments in the following areas were developed:

- Streamline Path Planning/Extensions
- Cooperative reconnaissance (ISR)
- RoboFlag system: adoption within community for basic, semi-autonomous research
- RoboFlag HitL Studies; initial modeling results
- Architecture for Evolution of Pre-planned Strategies and Resource Deployment using GP
- Team Tasking using tiered optimization

Detailed discussion of the results, methods to achieve the results, and detailed reporting in conference and journal publications are given in the following chapters.

Key limitations/gaps that remain include 1) systems level integration and testing, especially real time, 2) development with a "truly" intelligent adversary, and 3) development of a true understanding of how best to integrate a human operator into the loop. Future work mimics these areas, and includes: 1) Theory/studies on how best to enter the human element into the overall system architecture, 2) packet/Comm based control theory, as experimental evidence shows that approximately 10% of all packets can be lost in communications within cooperating vehicles, thus inhibiting performance, and 3) spatio-temporal and robust cooperation, such as low probability of detection, situational awareness mission, and 4) real time validation of the concepts in a systems level study.

CHAPTER 2: INTRODUCTION

Variable initiative control of automa-teams, or semi-autonomous battlefield management of multiple UAV's ground rovers and troops for example, is both an exciting prospect and an extremely complex challenge. By attempting to automate these complex battlefield (or similar) situations, the human could be freed from dull, dangerous, and costly tasks. In addition, the developing group could lead a new generation of military superiority. But, many complex technologies must be developed and integrated together, as well as with human operators. In addition, there are safety and security issues such as weapons release authority and operator training.

The Cornell led team developed a suite of technology tools, several human in the loop studies of operator control of multiple vehicles, a hardware simulator, and real time verification of many products. A summary of the Cornell led program and concepts is given in Ref. [1] (and printed in the appendices, as are all publications referenced within). The Cornell led MICA program was decomposed into a hierarchical architecture (Figure 1). As shown, the architecture addresses three core areas: Team Composition and Tasking, Team Dynamics and Strategizing, and Cooperative Path Planning. MICA also addresses Operator Interface in terms of information definition (rather than the physical interface, and Uncertainty Management ensures that each block performs in a realistic scenario. In addition, several DARPA (and other) programs were built upon, including the Software Enabled Control program and Honeywell's Playbook concepts. The work here addresses complexity (and novelty) of the MICA program from a *systems* perspective, i.e. from the top down.

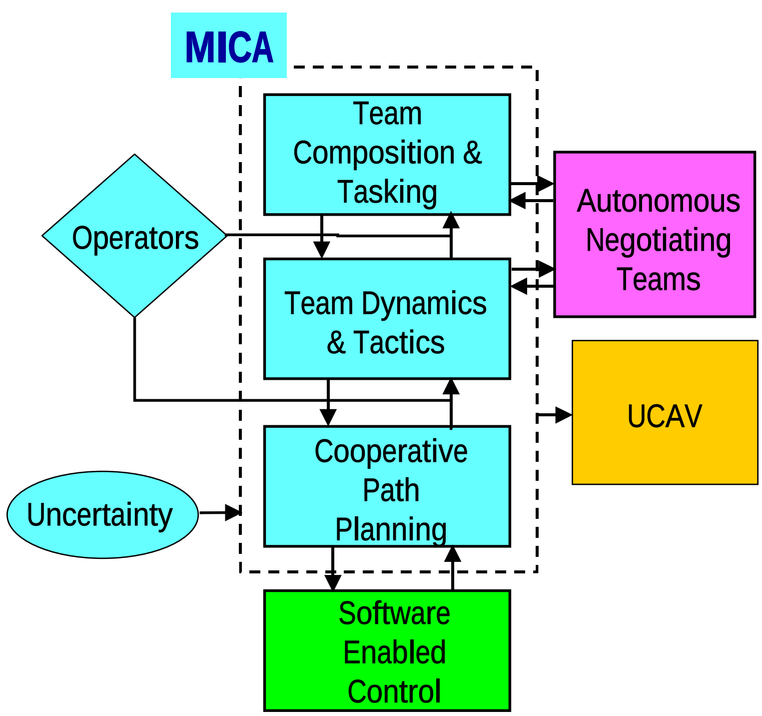


Figure 1: DARPA MICA hierarchical architecture.

The fundamental key question of the MICA program is: "How can M operators control N vehicles, where $N>>M$?" It is important to realize that the state of the art UCAV program has a 4 operator to 1 vehicle requirement in Phase I – far from the objectives of the MICA program. Even though the three PI's come from the technology background of the other blocks, it is our belief that the human information interface requirements *must* be defined first in order to understand how all of the underlying technologies will be integrated. It is a top-down approach to the problem (typical of systems approaches), rather than bottom up (from the technologies).

> The goal of our program was to develop, both theoretically and experimentally, an architecture for realistic semi-autonomous systems composed of human operator(s) and different (semi)autonomous vehicles using a variable initiative goal setting that achieves a core MICA objective: N vehicles >> M operators.

This work built upon two key concepts: 1) the human information interface (type, amount of information, etc.) must be defined first, and 2) a simple yet fairly complete demonstration can lead to faster, more complete solutions in the short term. The proposed experimental demonstration, called RoboFlag, is experimental testbed with autonomous, fast-moving teams of vehicles, and is therefore an excellent system to aid in the development and evaluation of realistic solutions to the MICA program. We used wheeled robots (analogous to ground vehicles and people), and floating vehicles (analogous to UAV's) to compete. The objective of the RoboFlag competition is to venture into opponent territory, locate and capture the "flag," and return with the flag back to the "home base." This has many key aspects to assess the objectives of the MICA program, including a human operator, team dynamics, different levels of tasking, cooperative planning, and uncertainties such as incomplete information, latency, intelligent adversary, neutral entities, etc. The environment also extremely dynamic, thus requiring a MICA type architecture. For this program, mobile vehicle testbeds at Cornell and Caltech were utilized.

Based on the program goal, the original specific objectives of the research are as follows:

OBJ 1. Define a human information interface (requirements) that meets the requirements of M=1 operators, and N=5 vehicles in the mixed initiative control setting.

OBJ 2. Define a human information interface (requirements) that meets the requirements of M=2+ operators, and N=10+ vehicles in the mixed initiative control setting, while codifying how the interface has scaled from **OBJ 1**.

OBJ 3. Develop/integrate technologies in each of the MICA blocks (Uncertainty Management, Team Composition and Strategizing, Cooperative Path Planning) to support the interfaces defined in **OBJ 1** and **OBJ 2**, and allow the development and comparison of mixed initiative control strategies.

OBJ 4. Demonstrate, in an experimental setting, a single N=5 on N=5 competition, as well as multiple competitions across a computer network (Cornell and Caltech), with all aspects of the environment being realistically uncertain.

OBJ 5. Work with industry, academia, and government partners to transition the technology within the MICA program, as well as to other applications.

The proposed program is focused on the Variable Initiative Block of the MICA architecture, as well as all interfaces to the other blocks. Because of the abrupt end of the program, progress was made in portions of OBJ 1,3-5 only.

CHAPTER 3: METHODS

The approach to achieving the proposed goals can be described in terms of work areas, as well as integration into the hierarchical structure. The work areas are shown in Figure 2 using four inter-related tasks: Human Interface development, (Human Centered) Mixed Initiative Control using M=1 and M>1 operators, Experimental Validation using RoboFlag Challenge Problems, and Technology Transition.

Human Interface

As stated previously, the first important step in this process was to define a set of human interface requirements for the MICA program in general, and RoboFlag in particular. This work was supported by the Cognitive Engineering team at Catholic University, SIFT, and AFRL. The plan had been to first define a set of interface requirements to address M=1 operators first, and eventually move on to M>1. In addition, the results were to be analyzed to understand if and how a scaling of the requirements and interface can be developed.

Mixed Initiative Control using M=1 and M>1 Operators

The second task was to develop Mixed Initiative control strategies. The focus for this task is on how a *human centered* system is designed for MICA type applications. Our work attempted to take a slice through the MICA paradigm, such that the full MICA hierarchy is used. This task developed algorithms for teaming that work under the constraints of M=1 and M>1. This task also leveraged technologies from other PI and Co-I programs (DARPA, AFOSR, NASA, etc.). In this task we also directly addressed uncertainty management, as it is such an important part of the final product and it is not addressed in the other PI's programs.

RoboFlag Challenge Problems

The third task was to develop the RoboFlag environment and use a series of challenge problems to validate the mixed initiative control strategies. Hardware from Caltech and Cornell was leveraged, making it much less expensive, and on-line very quickly (within the first six months). Subsequent sections show the hardware and MICA analogy for RoboFlag. A series of challenge problems were used to validate the real time implementation of the technologies and hierarchical structure.

Technology Transition

The fourth and final task for this program was Technology Transition. Specific items that our team will work on include: 1) integration of our algorithms into those of MICA partners that focus on the full hierarchy, 2) transition of our technologies to the open control platform, 3) integration of MICA partner's technology into the RoboFlag competition, and 4) development of the challenge problems. The primary focus of this transition was working with Alphatech, one of the industry leads of the program.

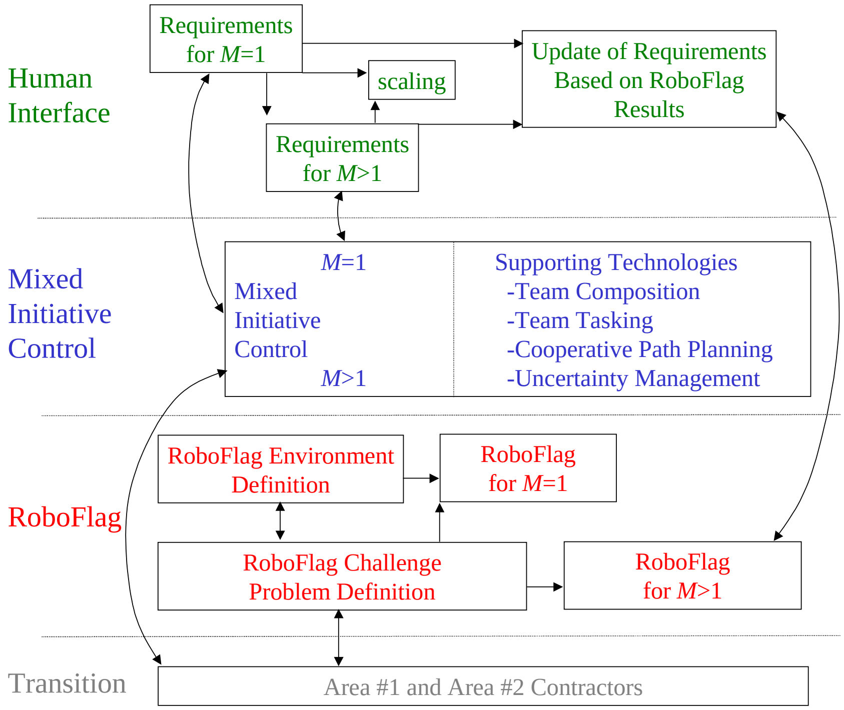


Figure 2: Conceptual outline of the proposed program showing four major tasks, interconnections, and approximate timeline.

Figure 3 shows a breakdown of the areas where the PI's and Co'I's worked. The percentages given indicate how much original effort was used in this program, as opposed to simply leveraging other programs/areas.

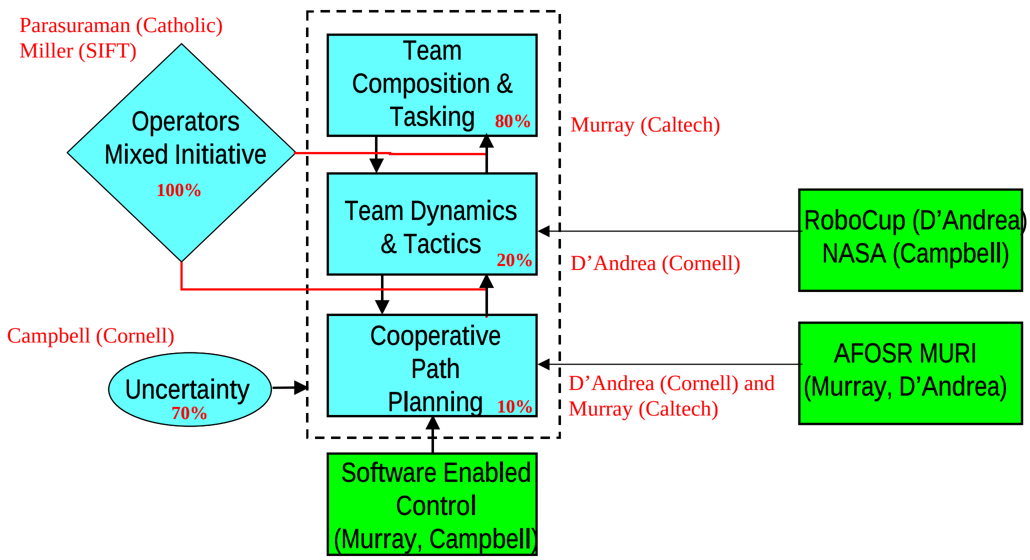


Figure 3: Proposed team breakdown mapped on to the MICA hierarchy. Red names and numbers indicate team leaders and *new* proposed effort. Green blocks indicate programs that will be leveraged in order to augment the effort each block.

This program explored the issues of the Operator (Human) interface in the context of the MICA architecture and Mixed Initiative Control in a realistic setting. With information requirements in mind, a hierarchy can begin to form, as shown in Figure 4. In the proposed approach, lower levels of the hierarchy, such as estimation and path planning, are defined by formal algorithms with hard guarantees. Mid-Levels of the hierarchy, such as team strategizing and composition, are defined by optimization and/or randomized algorithms that allow teams to perform operator tasks successfully, but not predictably. The operator interface occurs at multiple points in the hierarchy.

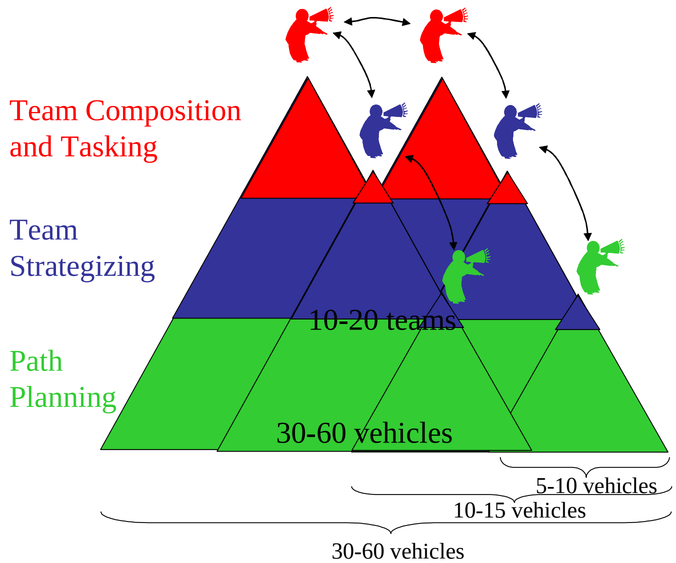


Figure 4: Proposed hierarchy and approach for the Cornell led team.

Because of the complex systems type setting of the MICA program, each block was integrated to fully benefit from the work in this area. In order to do this, we will take a full slice through the MICA hierarchy. The areas are described as follows:

Operators – This is an important area of the program, where we planned to address the human information requirements, decision making models, uncertainty issues, and other concepts. The complete operator interface (and associated technologies) were not defined (such as the display, etc.), the information content was to be explored.

Cooperative Path Planning – This area will see a low effort level, primarily because several of the PI's are working in this area under other (supporting) programs. The area explored the most was to be streamline path planners, building on theoretical work at Caltech and integration work at Cornell.

Team Composition – Several types of Mixed Initiative control strategies were to be developed, based on the requirements from the operator interface. Specifically, nontraditional concepts will be used such as operators controlling the "shape" of the automata, or other metrics such as center of mass. Another concept is called "Playbook," where the operator acts as a "quarterback" in a football game. He/she can call plays that are executed by the team over a short window of time afterwards. Based on feedback information, a new play is called.

Team Strategizing – Our work in this area focused on strategies developed using genetic programming. Resource allocation was also to be addressed, along with integration with the "Playbook" concepts.

Uncertainty Management – This area built on PI Campbell's DARPA SEC work, which includes a complex modeling strategy for nonlinear systems. This includes stochastic and/or hard uncertainty bounds, state estimates from noisy incomplete data, and multiple model integration (environment, aircraft, faults, etc.). The work in MICA extended the work to include cooperation, and integration with the concept of a "World State." This world state includes different levels of model fidelity for different levels of the hierarchy (entities health, resources (power, comm, etc.), adversaries, etc.) In addition, more difficult concepts will be addressed such as confidence factors required for each level, latency, and information outages.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Technological and functional accomplishments and achievements

During the 2+ years of the Cornell led program, accomplishments in the following areas were developed:

- Streamline Path Planning/Extensions
- Cooperative reconnaissance (ISR)
- RoboFlag system: adoption within community for basic, semi-autonomous research
- RoboFlag HitL Studies; initial modeling results
- Architecture for Evolution of Pre-planned Strategies and Resource Deployment using GP
- Team Tasking using tiered optimization

The following subsections details a summary of the basic theory, results, and significance, while the appendices give the full details of the work in the form of publications.

4.1.1 Streamline Path Planning and Extensions

Potential field methods offer a natural way for a user to interface with a group of vehicles. Rather than assuming direct control over vehicle behavior, a strategy which limits the number of vehicles an operator can control, the user shapes the world that the vehicles perceive by adding obstacles, goals, or other primitives. These primitives can then be composed into a resultant field which governs vehicle behavior and expresses operator intent; vehicles then perform low-level control tasks to which computers are well suited. If the operator is temporarily taken away from the control task, the vehicles have behavioral guidelines encoded in their perceived potential field that allow them to continue to behave in a desirable manner.

Ad-hoc methods for composing artificial potential fields frequently generate local minima which may trap vehicles in equilibria other than the goal state. As described in Ref. [2], a useful approach is to use a hydrodynamic concept of a stream function}, ψ, which satisfies Laplace's equation

$$\nabla^2\psi \triangleq \frac{\partial^2\psi}{\partial x_1^2} + \frac{\partial^2\psi}{\partial x_2^2} + \ldots + \frac{\partial^2\psi}{\partial x_n^2} = 0$$

and gives components u, v of fluid velocity in the xy plane

$$u = -\frac{\partial\psi}{\partial y}, \quad v = \frac{\partial\psi}{\partial x}.$$

The complex potential w of an irrotational two-dimensional flow of an inviscid liquid is defined by $w = \phi + i\,\psi$, where ϕ and ψ are the potential and stream functions defining the flow. One can assume for w any holomorphic function of z and the real and imaginary parts give the potential (gradient) and stream functions for a possible flow satisfying Laplace's equation. As solutions to Laplace's equation, stream functions (and their partner potential functions) have no local extrema and the flow must be tangent to obstacle surfaces, resulting in smooth paths.

The streamline concept is then extended to more complex planning approaches, including planning around multiple moving, uncertain obstacles, and integration into a multiple vehicle strategizing

approach. The insertion of an obstacle into a flow introduces the boundary condition that the flow be tangent to the surface. A useful historic result here is the Circle Theorem, which gives the complex potential *w* resulting from placing a circular obstacle of radius *a* at the point $b = x_0 + i\, y_0$ in a flow with complex potential *f(z):*

$$w = f(z) + \overline{f}\left(\frac{a^2}{z-b} + \bar{b}\right)$$

The Circle Theorem allows the stream function to be composed of primitives which describe different vehicle behaviors. The two most useful primitives are the sink, $f(z) = -C\ ln(z)$, and the vortex, $f(z) = C\ i\ ln(z)$, where *C* is the strength of the singularity. In practice, *C* is arbitrary and the velocity is normalized to the vehicle dynamics while preserving its direction. \fig{streamlines} depicts the streamlines obtained for a vortex and a sink flow with an obstacle. The paths generated by following the streamlines tend to be smooth, and therefore at least qualitatively well-suited to the dynamics of an aircraft-like vehicle. This is shown in the figure below.

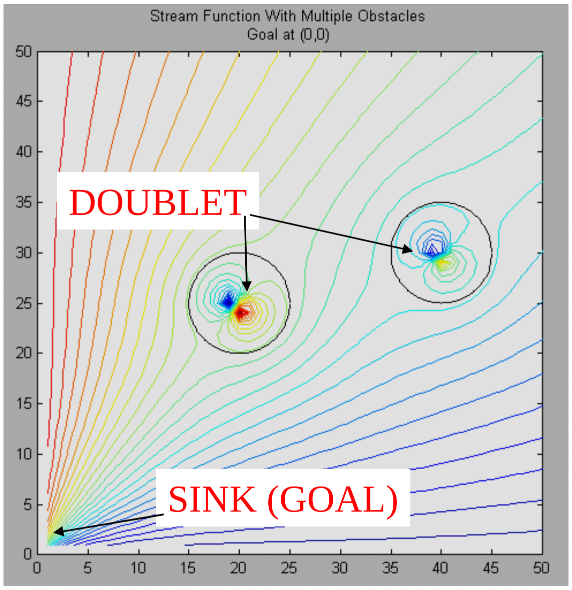


Figure 5: Streamline theory for path planning: Doublets are used to define an obstacle, and the streamline theory allows smooth paths from the course to the sink.

If the obstacles to be avoided are moving, generating the stream function in a quasi-static manner is insufficient to guarantee obstacle avoidance in a dynamic environment. The correct boundary condition becomes that the vector field must be exterior (or tangent) directed on the boundary of the obstacle in the rest frame of the obstacle. Stream functions offer a convenient method for handling this condition.

If the obstacle from the Circle Theorem above is moving with constant velocity $\overline{v}_0 = v_x + iv_y$, the complex potential for the flow about the obstacle is given by:

$$w(z) = w_s(z) - v_x\left(\frac{a^2}{z-b}+\bar{b}\right) - iv_y\left(\frac{a^2}{z-b}+\bar{b}\right)$$

where w_s is the static stream function that would be derived if the obstacle were not moving. Please see Ref. [2] for a more thorough description of the streamline theory, and Ref. [3] for additional multiple vehicle planning approaches with multiple dynamic obstacles using streamlines.

In summary, the streamline path planner and its extensions have the following benefits:

- Smooth, aircraft like trajectories
- Multiple, moving obstacles
- Risk based planning
- Guarantees
- Fast computation
- Synthesis allows complex behavior

All levels of the control hierarchy - path planners, operators, vehicle control systems - require estimates of information based on sensed data. Traditional approaches include Kalman Filtering and its many derivatives. The approach here is to develop a formal bounded estimation architecture for multiple vehicles that enables path planning in an uncertain, realistic environment. Adversarial vehicle positions and uncertainties are tracked using traditional radar sensing and a bounded set membership filter (ESMF), while cooperative vehicle positions are tracked using GPS type sensing and communication cross-links. Uncertainties addressed specifically in the architecture include model errors, model nonlinearities, sensor noise, and radar and communication black-outs.

The ESMF delivers an "ellipsoid set'" at each time step, where the state may lie within an uncertainty ellipsoid. The ellipsoid is analogous to the covariance of the KF and EKF, but here, the ellipsoid bounds the probability of the estimate error. The bounded ellipsoid set is also an excellent choice for use within path planning approaches, as most of these methods consider circles and ellipsoids as obstacles. So, the process here is that the filter delivers a set of bounded probability obstacles, and the streamline path planner is used to plan paths around these obstacles.

The ESMF is developed as follows. Assuming a general form for a model of the vehicle/system being estimated,

$$\begin{aligned} x_{k+1} &= f(x_k) + w_k \\ y_{k+1} &= g(x_{k+1}) + v_{k+1} \end{aligned}$$

the initial state is assumed to be bounded at a given level of probability using an ellipsoid:

$$\mathbf{x}_i(0) \in S(\hat{\mathbf{x}}_i(0), P_{i,j}(0)) \quad at\ \mathcal{P}_{i,j}$$

where the ellipsoid is defined as:

$$[\mathbf{x}_i(0) - \hat{\mathbf{x}}_i(0)]^T P_{i,j}(0)^{-1}[\mathbf{x}_i(0) - \hat{\mathbf{x}}_i(0)] \leq 1$$

The prediction step predicts the evolution of the initial set over a small time horizon. Linearizing about the nonlinear model about the current center of the state uncertainty ellipsoid yields

$$x_{k+1} = f(x_k)\big|_{\hat{x}_k} + \left.\frac{\partial f(x_k,t_k)}{\partial x}\right|_{\hat{x}_k} \Delta x_k + \text{H.O.T.} + w_k$$

where H.O.T. refers to the higher order terms of the expansion. The uncertainty ellipsoid in x_{k+1} is developed by evolving ellipsoids from the previous state (x_k), higher order terms (H.O.T.), and disturbance (w_k), and adding them together. The traditional EKF ignores the higher order terms, whereas this is a key step in the ESMF (using the using the Taylor residual) that allows formal stability of the algorithm to be developed. Therefore, each obstacle is propagated forward in time for each probability level. This is given as:

$$
\begin{aligned}
\hat{\mathbf{x}}_i(k+1) &= f(\hat{\mathbf{x}}_i(k)) + \hat{\mathbf{u}}_i(k) \\
P_{i,j}(k+1) &= A_k \frac{P_{i,j}(k)}{1-\beta_j} A_k^T + \frac{\hat{U}_{i,j}}{\beta_j}
\end{aligned}
$$

where

$$
A_k = \left.\frac{\partial f(\mathbf{x}_k)}{\partial x}\right|_{\mathbf{x}_k = \hat{\mathbf{x}}_i(k)}
$$

The update step, which uses a measurement and the output equation of the model, is developed similarly. The Update Step is an intersection of two sets (the predicted state and the projected state from the sensors). The ESMF can then be used in subsequent applications such as adversary detection and information fusion with limited communications.

The next step is to integrate the obstacles with the streamline path planner. Figure 6 shows an example of how this bounded probability filter is used conceptually. Here, a path planner must plan from a start to an end point, with obstacles along the way. A low risk path can be defined by grouping overlapping targets, and moving on a path around the grouped obstacles. A high risk path can be defined by maneuvering through the targets using ellipsoids with a smaller probability of enclosure.

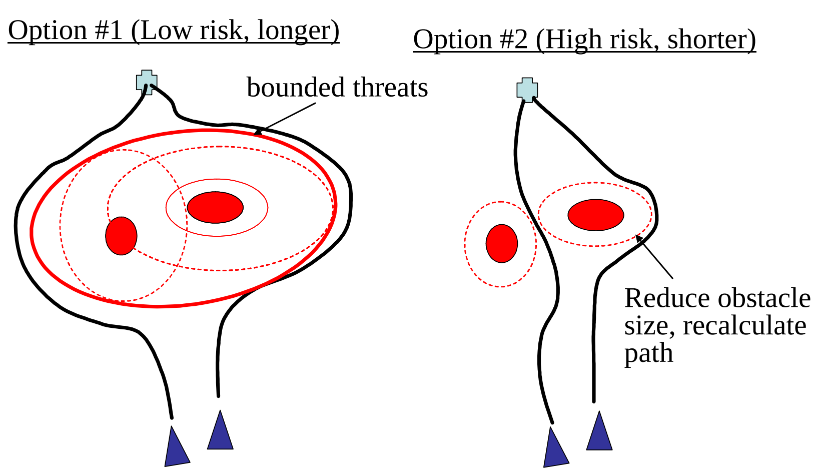


Figure 6: Example of a streamline path planner integrated with the bounded probability estimator. High risk paths around smaller obstacles, and lower risk paths around larger obstacles, can be developed.

A second extension developed was to add time into the path planner. In this case, each path was assumed to be at a constant velocity. Using this information, the time for each path could be approximated, and used to integrate path segments together. An example of this is shown in the figure below.

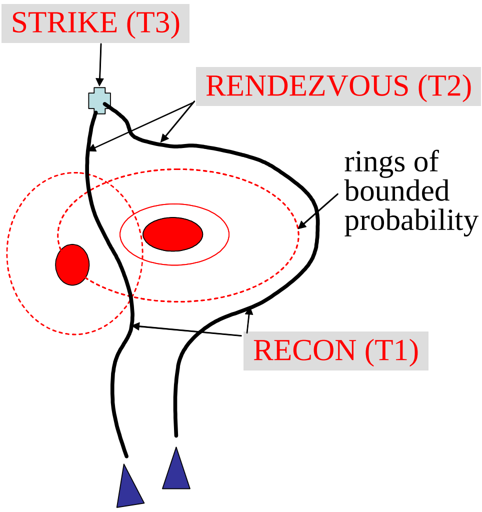


Figure 7: Example of a streamline path planner that integrates several paths together, and adds time constraints (T1-T3).

All of the work developed was also extended to be completed integrated into the OEP (Figure 8-Figure 9). In addition, the work was integrated with Alphatech's OEP implementation, demonstrating a transition of the technology (Figure 10).

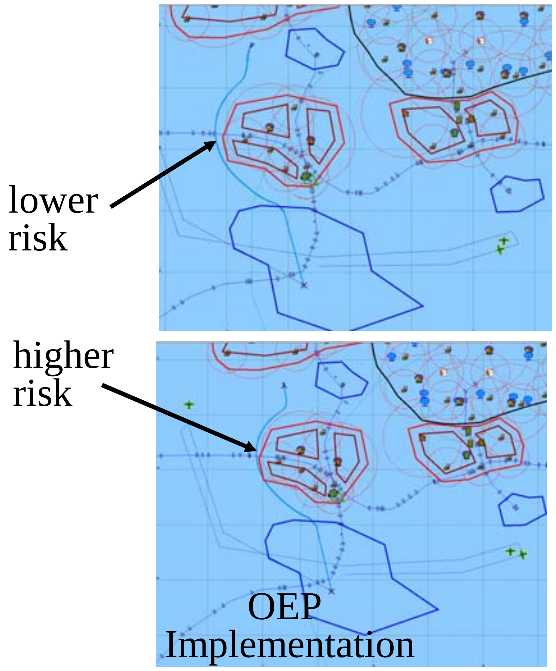


Figure 8: Example of the streamline path planner and bounded probability estimator working in the OEP.

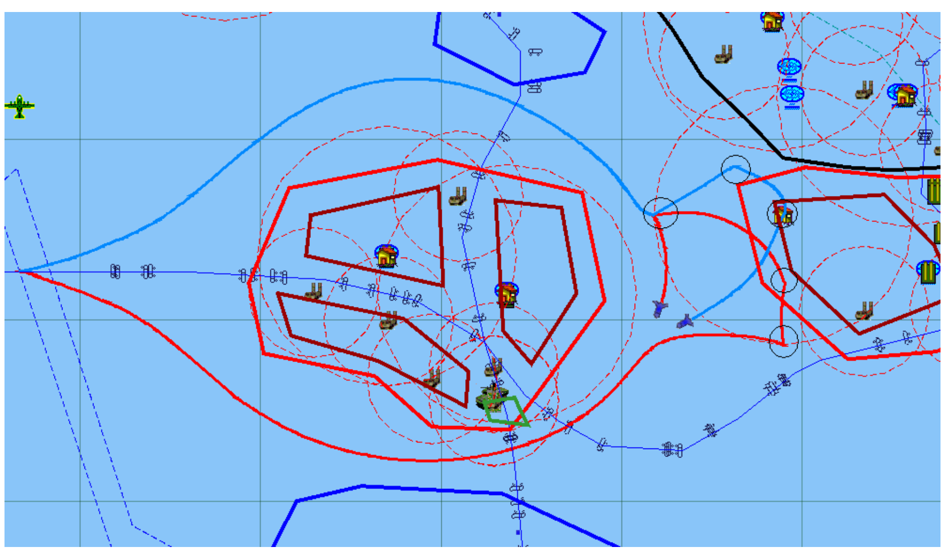


Figure 9: Example of the streamline path planner, bounded probability estimator, and time constraints working in the OEP.

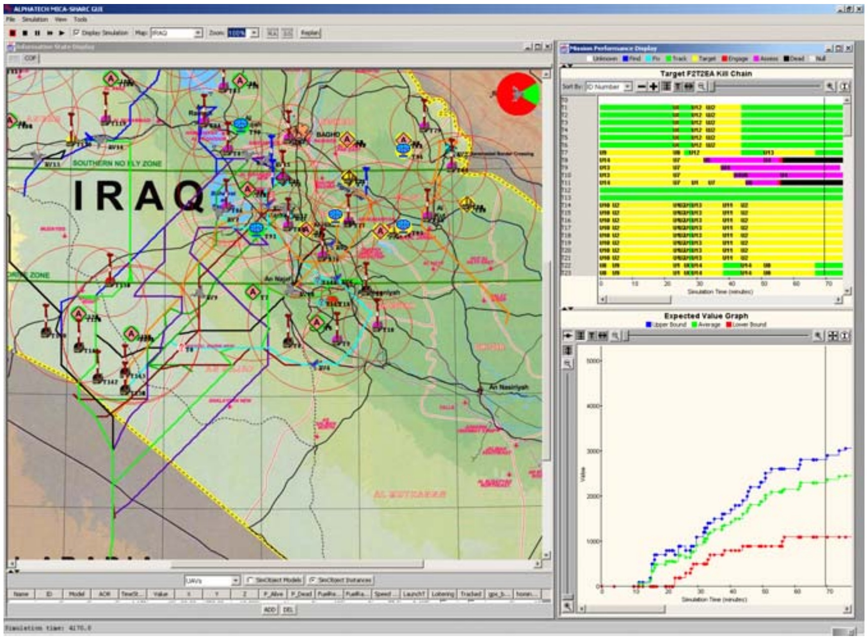


Figure 10: Demonstration of the streamline path planner in Alphtech's OEP.

4.1.2 Cooperative Reconnaissance

There are distinct advantages to having more vehicles in the application domain, with increased performance being a primary advantage. A good example is reconnaissance, which can be accomplished more quickly and reliability using a larger number of vehicles. In addition, conservatism of the target locations is decreased; this allows more precise offensive plans to be developed, including path planning.

The work proposed here focuses on coordinating multiple dynamic sensors, typically on moving vehicles, in order to attain the best estimates of the environment as possible. The work is at the boundaries of state of the art in estimation and task planning areas. As a motivating example, consider the dynamic search problem shown Figure 11. The characteristics of the problem include:

- There are N vehicles, each with position sensors with different capabilities. Each vehicle also has dynamic constraints, such as max/min velocities and min turn radius.
- There are M targets distributed throughout a given area; a subset of the targets are dynamic.
- The vehicles have T time to search an area and collect position information on all targets.
- There are areas around each target and within the search zone that are stay out zones.

Given these characteristics, the problem is to plan task assignments (and indirectly path trajectories) for each vehicle with the objective of minimizing uncertainty in the position estimates. Said another way, the vehicles plan where they must cooperatively move in order to acquire as much *relevant* information about their targets (and therefore minimize uncertainty) as possible. This complex problem raises several important questions that must be answered at the basic level:

- Does the implementation scale well as N, M increase?
- What happens if one of the sensing vehicles has a fault?

- Can task assignments be re-planned quickly if target priorities change?
- How does the solution change if there are communication constraints in area (max radius, bandwidth) and networking (max links)?

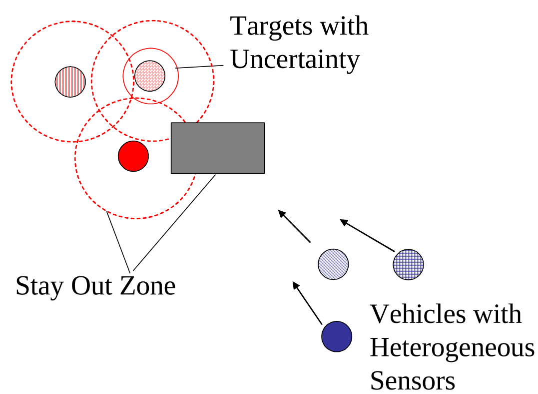


Figure 11: A motivating example for cooperative information seeking and planning.

In order to illustrate several of the important aspects of the concept proposed here, consider the case where there are N = 2 vehicles are cooperating in order to collect information on the environment, and M = 3 targets. One typical solution is outlined in Figure 12 as three steps. Step 1 defines a set of primitives for each vehicle; these are typically point to point maneuvers about areas of risk, within the vehicles constraints, etc. The set of points are defined as possible areas of interest to explore – areas with large information yield in this case. Step 2 assembles each of the primitives, along with other items such as path risk, total time required, time constraints, etc. into a table of options. It is important to realize that Steps 1 and 2 are completed very fast because close form analytical solutions are used. Step 3 then uses integer programming to solve for the best option(s), thus defining tasks for each vehicle.

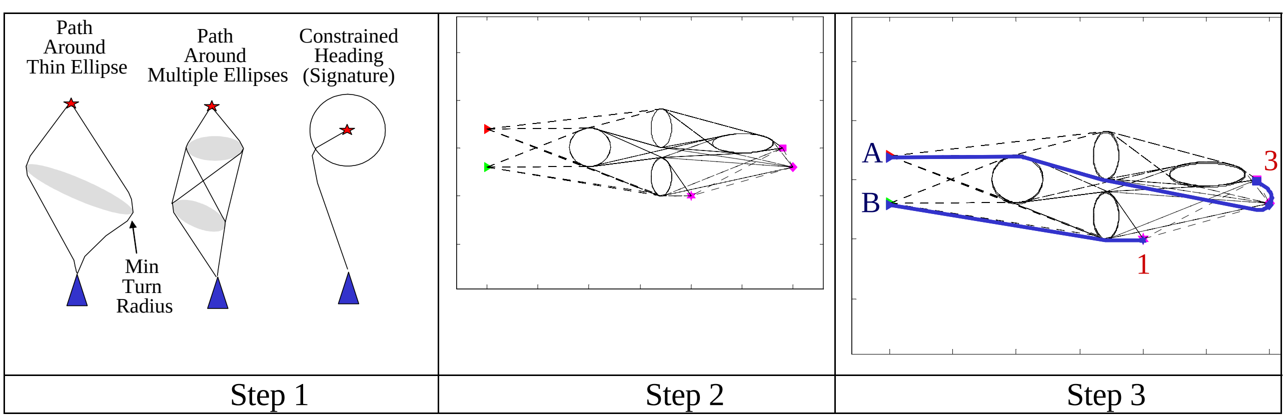
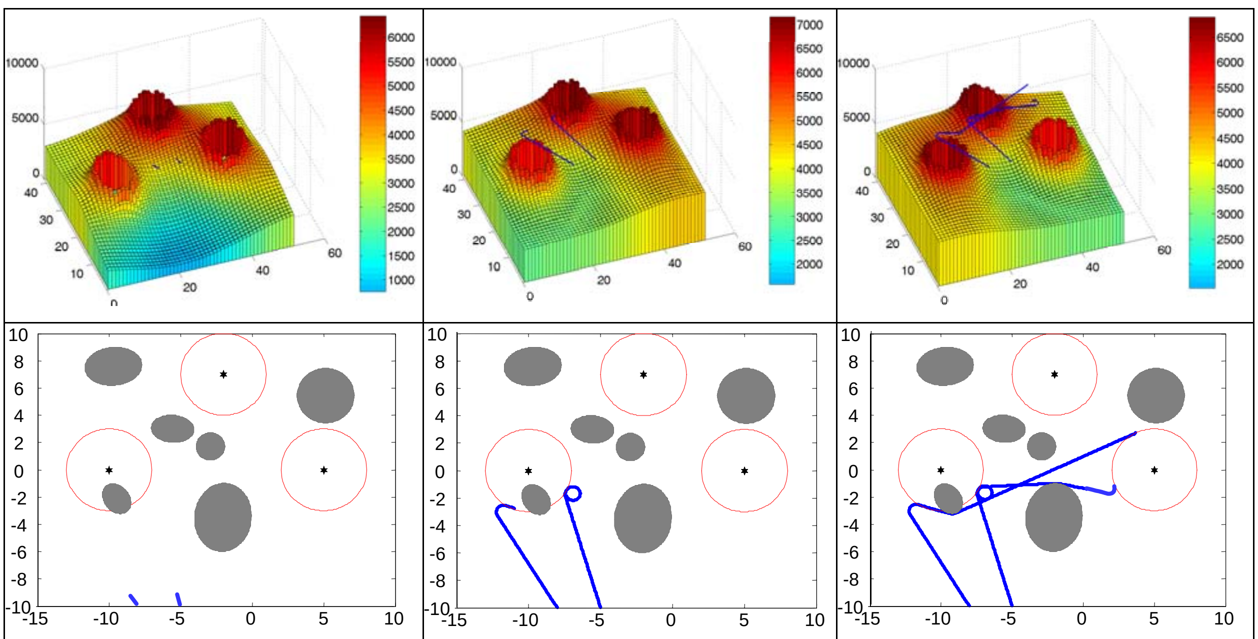


Figure 12: Three steps to the cooperative reconnaissance. Step 1: Use vehicle primitives to quickly plan paths from one point to the next. Step 2: List out all options, including information ability, total time, time constraints, risk, etc. Step 3: Select path/task.

Applying this concept to cooperative estimation, consider the example shown in Figure 13. The example includes $N = 2$ vehicles, each with vehicle constraints on speed (min and max) and turning radius. Each vehicle also has a radar type sensor with ellipsoidal uncertainty bounds. There are $M = 3$ targets, each with a circular stay out zone of a given radius. There are six neutral areas (in this case ellipsoids but they could be any polygon also) which are also considered stay out zones. The objective is to collect as much position "information" as possible on all three targets using the sensors on the two vehicles (equivalent to minimizing the location uncertainty). Constraints added to the problem include vehicle, stay out zone and maximum time constraints.



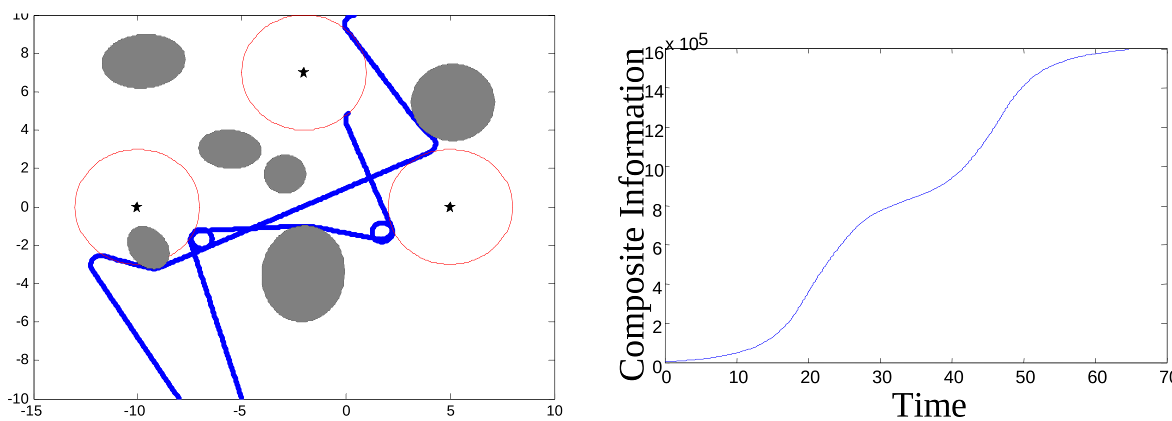
(a) (b) (c)

Figure 13: Example showing initial results. Top row shows the combined information as a function of X-Y position, given the location of one vehicle. The bottom row shows the path of both vehicles. (a) Initially, the vehicles are far away from the targets and information yield is smaller. (b) At the first target, the vehicles triangulate, producing high information yield. (c) Triangulation occurs at the second target, again producing a high information yield.

Based on the 3D information maps, it is deduced that the best information yield is to move as close as possible to each of the targets, i.e. up to the outer radius constraint. Therefore, the points defined for the vehicle primitives were 8 points about each of the target radii. Therefore, the integer programming problem effectively optimized which points around each target the vehicles should move towards. The results shown in Figure 13 demonstrate that

1. It is most beneficial to move close to the target and triangulate. The "smart: use of multiple points works well in recovering the optimal configurations
2. The fastest approach without obstacles is very similar to the minimum spanning tree with the shortest path from target to target
3. As the required time decreases, the vehicles tend not to triangulate around each target, but to send one vehicle to each target and triangulate from a larger distance along the away.

Figure 14(a) shows the final path of the two vehicles. Notice that around each target there are specific points to where each vehicle moves. These points attempt to triangulate around each target, which is intuitively correct: Triangulating using radar sensors increases the "information" on the sensed target. It is also noted that the time constraints are added to the optimization in order to require that each vehicle triangulate around a given target at the same time. Figure 14(b) shows a measure of the total information collected over time for the problem. Notice that as the vehicles move closer towards a target, the information increases (both because of proximity and because of triangulation).



(a) Final path option, showing triangularization about each target.	(b) Composite information for both vehicles over time.

Figure 14: Example of cooperative estimation.

A key step in the work here is the definition of cooperative information. The work here will utilize an information form of an estimation filter. Given a state estimate $\hat{x}_{k,k}$ and state error covariance $P_{k,k}$, the information state $\hat{\hat{x}}_{k,k}$ and information matrix $I_{k,k}$ are defined as:

$$\hat{\hat{x}}_{k,k} = P_{k,k}^{-1}\hat{x}_{k,k}, I_{k,k} = P_{k,k}^{-1} \tag{1}$$

As each measurement is added to the system, more "information" is collected. One measure of the ability of a sensor to add information is the Fisher Information Matrix (FIM), defined as

$$FIM = E\left[\nabla^T \ln\left(p(Z^k \mid \hat{x}_{k,k})\right)\nabla \ln\left(p(Z^k \mid \hat{x}_{k,k})\right)\right] \quad (2)$$

where Z^k is a batch measurement up to time k. Scalar cooperative measures of information for the ith target can then subsequently be defined as

$$\mathrm{I}_i = \left|\sum_{j=1}^{N} FIM_j\right| \quad (3)$$

And the composite information for M targets is then given as

$$\mathrm{I} = \sum_{i=1}^{M} \mathrm{I}_i \quad (4)$$

This scalar function works well for giving a measure of how well a sensor (or sensors) can cooperate. But, it does not scale particularly well because I (which requires solving M determinants) is a strong function of three variables: the time horizon used in the calculation for the *FIM's*, and the *X-Y* position placement of the N sensor placements. The work proposed here attempted to overcome these deficiencies by optimizing the placement of the "high information" points.

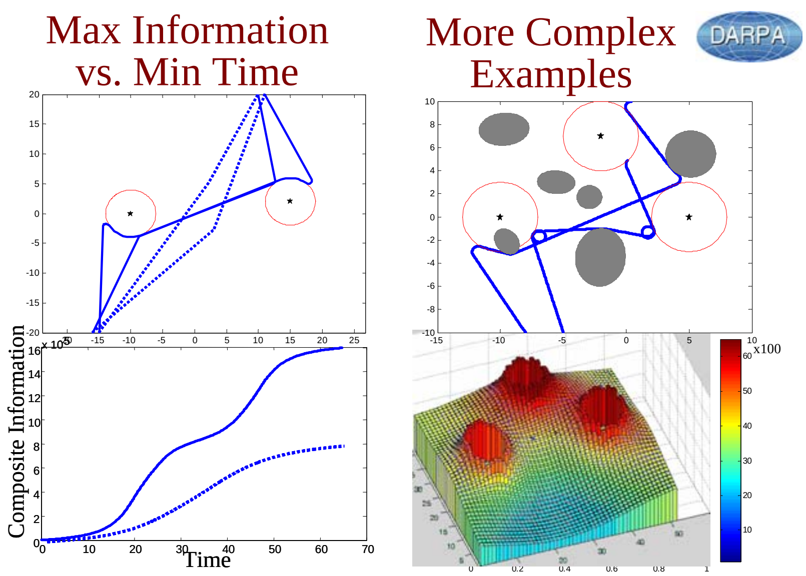


Figure 15: Additional examples showing the trade-off between maximizing information vs. minimizing time (left), and a more complex example with three targets, stay out zones, and a variety of obstacles.

4.1.3 RoboFlag System: In House Experiment

RoboFlag, is experimental testbed with autonomous, fast-moving teams of vehicles, and is therefore an excellent system to aid in the development and evaluation of realistic solutions to the

MICA program. We used wheeled robots (analogous to ground vehicles and people), and floating vehicles (analogous to UAV's) to compete. The objective of the RoboFlag competition is to venture into opponent territory, locate and capture the "flag," and return with the flag back to the "home base." This has many key aspects to assess the objectives of the MICA program, including a human operator, team dynamics, different levels of tasking, cooperative planning, and uncertainties such as incomplete information, latency, intelligent adversary, neutral entities, etc. The environment also extremely dynamic, thus requiring a MICA type architecture.

Figure 16 gives an overview of the RoboFlag hardware. Important components include:

- Robot entices, up to 11 on each side
- Two overhead cameras, taking pictures of two sides of the field. Each robot has colored dots on the top, which are used to track each robot.
- One vision computer, which includes all software used to track each robot's position and rotation.
- An arbiter computer, which is used to disseminate information, add uncertainty, and other aspects.
- Computers for each entity; a bank of computers used to house all algorithm/software for the vehicles.
- An RF transceiver system used to communicate commands to each robot.

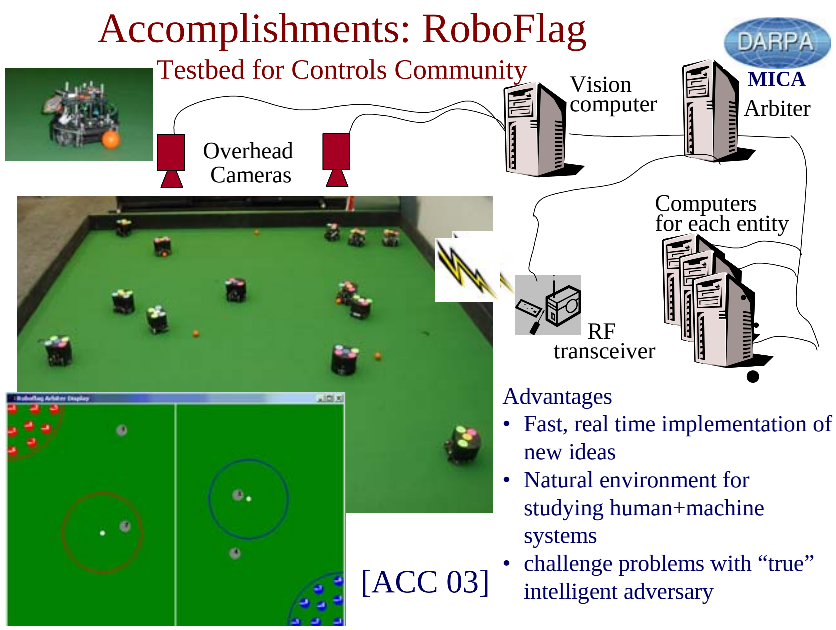


Figure 16: Overview of the RoboFlag system.

Figure 17: Four views of a typical RoboFlag game. Upper left: a video of the hardware running. Lower left: an arbiter view of the game, which represents all aspects of the system. Upper Right: An offense (blue) view of the game; note that blue can only see its own team now, based on its smaller sensor radii. Lower Right: An defense (red) view of the game; note that red can only see its own team now, based on its smaller sensor radii.

Figure 17 gives a summary of a small RoboFlag game, with four views of the game. In one view, there is a video of the actual hardware running. The arbiter view of the game shows all robots and their motions. An offense (blue) view of the game includes all blue robots, but not red robots. This is a result of the use of smaller sensor radii. Similarly, the defense (red) view of the game can only see the red robots. The reader is pointed to Refs. [6]-[11]

The following milestones were completed in the 2+ years of the Cornell led MICA program:

- Jan 2002: hardware complete
- April 2002: initial play demonstration
- July 2002: initial competition (SURF students) several plays working
- Oct 2002: 1:5 competition (Phase I)
- April 2003: 1:8 (teaming strategies – Phase II)
- July 2003: 2:10 (operator in hierarchy, heterogeneous vehicles – Phase II)
- August 2003: 2:10 (SURF 2003 – Phase II)
- Oct 2003: Full physical game (and simulation) of 2:10 at mid-point of MICA program

A variety of RoboFlag games (challenge problems) have been played, both on the simulator and the hardware. Examples include: 1) six on six vehicles (one operator on each side), 2) single vs. multiple operators, 3) offense vs. defense, 4) varied vehicle numbers and speed. A summary of these game results, and conclusions, are given in Ref. [12]. Figure 18 shows a view of two of the GUI's that have been developed for two evolutions of the game. In Figure 18(left), the GUI is based on path planning, and point and click commanding. In Figure 18(right), the GUI uses higher level plays, and heterogeneous vehicles.

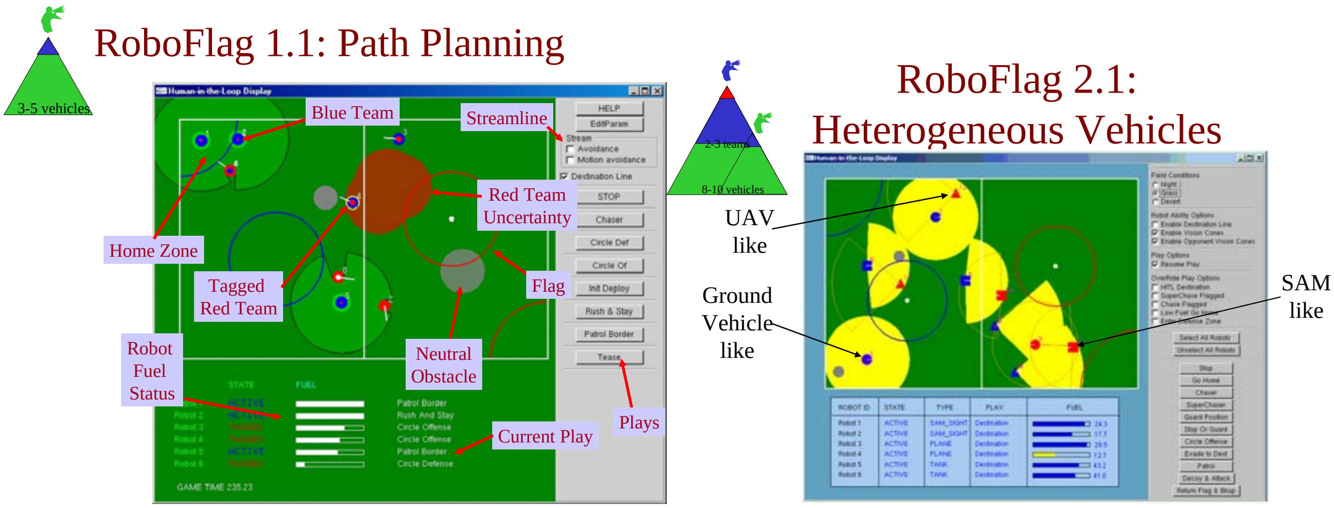


Figure 18: RoboFlag GUI's based on the evolution of the game.

Figure 19 shows an example of an offense vs. defense set of plays implemented on both the simulator and hardware. The goal of the offense is to capture the flag, while the goal of the defense is to prevent the flag from being taken. The defensive strategy is simply to straddle the defense zone, and chase offensive vehicles away. The offense works cooperatively to try to gain the flag by using a strategy: 1) two friendly vehicles move towards the adversary zone in tandem, 2a) friendly 1 moves in, appearing as it is moving towards the flag, 2b) adversary 1 moves towards friendly 1, creating a small amount of space on its other side, 2c) friendly 2 sneaks through to get the flag, 2d) friendly 1 moves out quickly without being caught, 3) steps 1-2,4 are repeated with adversary 2 to allow friendly 2 to move out of the defense zone.

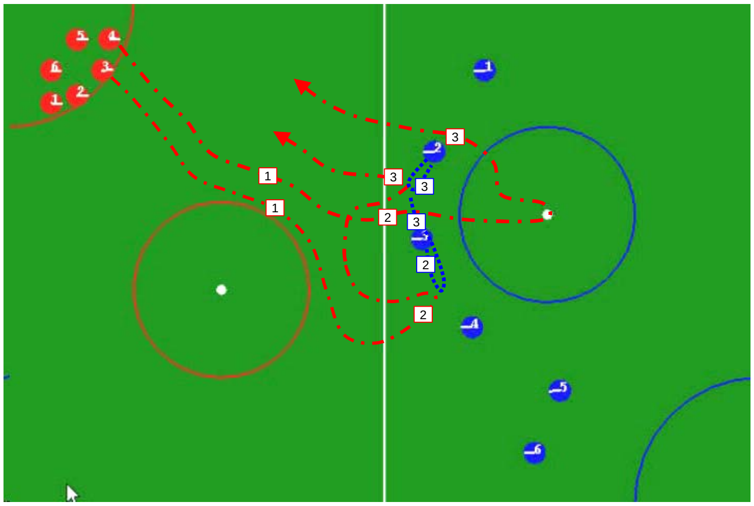


Figure 19: Simulation example of a play in RoboFlag.

4.1.4 RoboFlag HitL Studies

A scalable technique to model decisions and define interface information requirements is desired, which can aid in estimating automation level and cognitive requirements for resource allocation, play choice and strategy. The approach must be general so that user definitions at multiple levels of the hierarchy can be developed. The approach here assumes that operators are limited capacity information processors and can interpret and react to only a finite amount of information at a given time. Cognitive abilities and limits vary widely; the goal here is to define an average that can be used to develop a set of models. A driving question then becomes, ``How can automation be used as efficiently as possible?" More precisely, what type of interface design allows N>>M, while not over burdening (too much information, too fast), or under burdening the user (too little information, too slow). The latter situation is important in the case of UAV's and other heavily automated systems (nuclear power, air traffic control, etc.), where the user must maintain a high level of situation awareness.

An analytical approach to modeling the decision process has been developed, with human in the loop (HITL) testing on RoboFlag used later for validation. First, various plays and scenarios in the RoboFlag game are identified. For each, the inputs and outputs from user's perspective are enumerated and described. Simple cognitive models are used to define requirements. Higher level models are then developed by 1) defining a probability that an outcome will occur, and 2) using Markov Decision Processes (MDP's) to model the decisions within the play. These models can be built into a hierarchy and used to study decisions and their affect within the overall control hierarchy. An end goal is that automation can string plays together in order to maximize probability of certain outcomes as based on models of both the automation and operator decision processes.

As an example, consider Figure 20, where a ``reconnaissance play" has been defined. This play assigns several robots to venture into adversary territory and detect and locate adversary assets, and produce a probability map of adversary location and uncertainties. The operator decision process for this play is modeled using a flow chart (Figure 20) with four decision making blocks: acquire new information, analyze it, decide what to do and execute a command. An initial time model of this decision process has been developed based on simple models of human actions. Examples include the average time it takes to a) decide between two choices, b) point and click a mouse, c) recall a previous decision from memory, etc. Actions of the full decision process are decomposed into these time based models. Based on the reconnaissance play shown in Figure 20, a time of T_R = 3.72 sec is defined as the total time required to move through the full chart (Figure 21). The number can be compared to an upper acceptable bound for T_R. As an example, consider a robot with vision radius R and lateral speed s. A time requirement of $T_R < R/s$ is required to allow for user control. In the case of N robots that are supervised individually, it follows that $N\ T_R < R/s$. Obviously as the speed and number of robots increase, the ability of an human operator to control these vehicles violates the requirement.

In a similar fashion, models for other plays are developed. Ultimately, optimal design parameters and improved automation techniques are generated. These micro, and then synthesized, models of operator decisions are then integrated into a larger MDP model of the operator decisions - a model that can be used for prediction purposes. Unfortunately the program ended before the Cornell led modeling effort could be developed. It is noted that the SIFT team did do some initial MDP modeling in the program. However, the SIFT team decided to not accept the final dollars in the contract to finish off the final report, primarily because the work was still in its infancy.

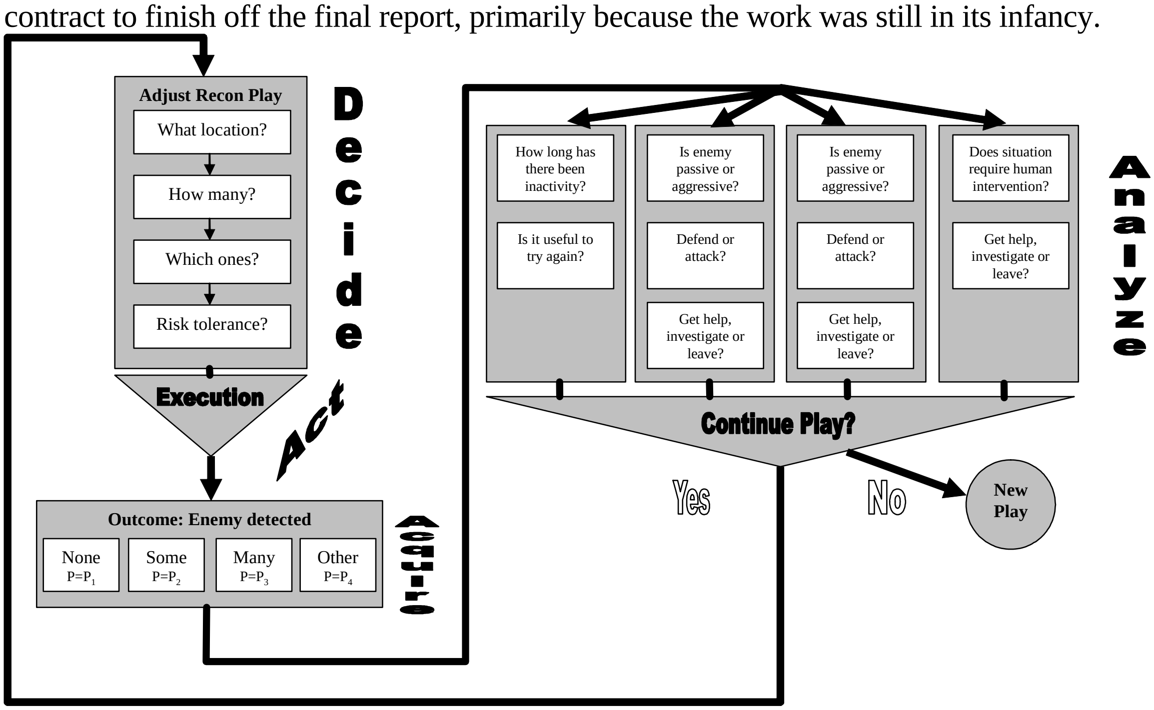


Figure 20: A simple model of a reconnaissance play, and the human operator decision process.

Example:	Single friendly robot (n=1) engaged in reconnaissance play. Single Enemy robot detected (outcome #2 from previous chart). User acknowledges. Commands friendly robot to continue investigating alone while defending.		
		t(seconds)	**comments**
Acquire:	Respond to audio cue (beep)	0.29	Respond to 3.4 audio cues/second
	Visually acquire enemy robot	0.17	Single target
Analyze:	Visually inspect enemy robot	0.34	Single target
	Does robot appear agressive? (n)	0.24	Two choices
	Attack or defend? (d)	0.30	Three choices
	Get help, continue observation or leave? (o)	0.30	Three choices
Decide:	Recall: Continue or leave	0.12	Recall of one item from short term memory
	Visually acquire part of map to be scanned	0.17	Single target
	Select that part	0.31	Fitts Law (d=4, w=0.5)
	Recall: Get help? (n)	0.12	Recall of one item from short term memory
	Make cognitive decision to choose number of robots (0)	0.30	Three choices: none, some, many
	Recall: Attack or defend? (d)	0.12	Recall of one item from short term memory
	Select attack/defend	0.22	Fitts Law (d=2, w=0.5)
	Make cognitive decision to choose risk tolerance mode	0.30	From est. 3 choices
	Select risk tolerance	0.22	Fitts Law (d=2, w=0.5)
Act:	Execution - click "ok"	0.22	Fitts Law (d=2, w=0.5)
Total:		**3.72**	

Figure 21: Sample of the interface time requirements developed for M=1, N=8.

The Cornell led team developed a series of RoboFlag games used to evaluate the information interface as a function of several important parameters, including the speed of the vehicles, the number of vehicles, the number of operators, etc. Logger data was first examined for trends in total score, number of mouse clicks, automation used, etc. A summary of these experiments are given in Ref. [12]. Phase I of the experiment consisted of ten sets of games. Each set contained 12 games of 400 game seconds (approximately 10 minutes) each. Nine of the ten games sets varied two parameters with one versus one human operator.

Game Parameters: Game speed and number of robots per a team were varied. Game speed was chosen so as to be correlated with increasing levels of user workload. Game speed was varied between three values: 0.25, 0.50 and 0.75 m/s.

Number of robots: Number of robots per team was chosen as a second parameter correlated with an increase in cognitive workload. Number of robots was varied between three values: two, four, and six per side.

Performance Measures: Game performance, user workload and situation awareness were assessed using both quantitative and qualitative methods. Surveys were used (TLX and SART) to qualitatively measure user's situation awareness, frustration level and cognitive workload.18,19 Users were also given an opened-ended questionnaire section for comments that allowed them to describe in their own words their experience after each game set.

Timeline and Procedure: Each game set contained 12 individual ten-minute games, with three occurring in parallel, or 40 total minutes of game play. Two game sets were run per testing day. This required six total days of data collection. The logger recorded data from each game. A total of 116 log files were generated. Participants played a game set and completed a TLX and SART

survey at the end of each game set. After taking a 15-minute break they played the next game set. Phase I data collection was completed by December 16th, 2002.

A summary of the parameters and number of games is given below.

	2 Robots	4 Robots	6 Robots
0.25 m/s	Set 1	Set 2	Set 3
0.50 m/s	Set 4	Set 5	Set 6
0.75 m/s	Set 7	Set 8	Set 9

Each set contains 12 games

	6 Robots (2v2 Operator)
0.50 m/s	Set 10

Set 10 contained 8 games

Figure 22: Phase I Experimental Parameters.

As a sample of the results, Figure 23 shows a plot of the average score and 1σ bound vs. time for games with one operator against one operator, and two operators against two operators. In the two operator games, several strategies were employed including splitting up the vehicles used between the operators, and having one operator look for strategies while the other implemented them. In either case, two operators usually performed better. Intuitively, this is correct. As shown in Ref. [12], this is the result of the information interface being at a more efficient level (i.e. the information interface has changed), thus allowing the performance to increase.

Several other figures are also given to present typical results of the study. Figure 24 gives a summary of the total score between Phase I games, and a second set of Phase II games that included several automated plays including circle offense and circle defense. Note that the score is consistently higher statistically. Figure 25 shows the automation employed versus number of robots at 0.50 m/s for one on one operator games. Note that several automation plays, such as the fuel override, are used when more vehicles are controlled. In Figure 26, the total score for two vs. two players as a function of software and hardware implementation is shown. Notice that the simulator results are higher; this is a function of the robots occasionally getting caught between camera views.

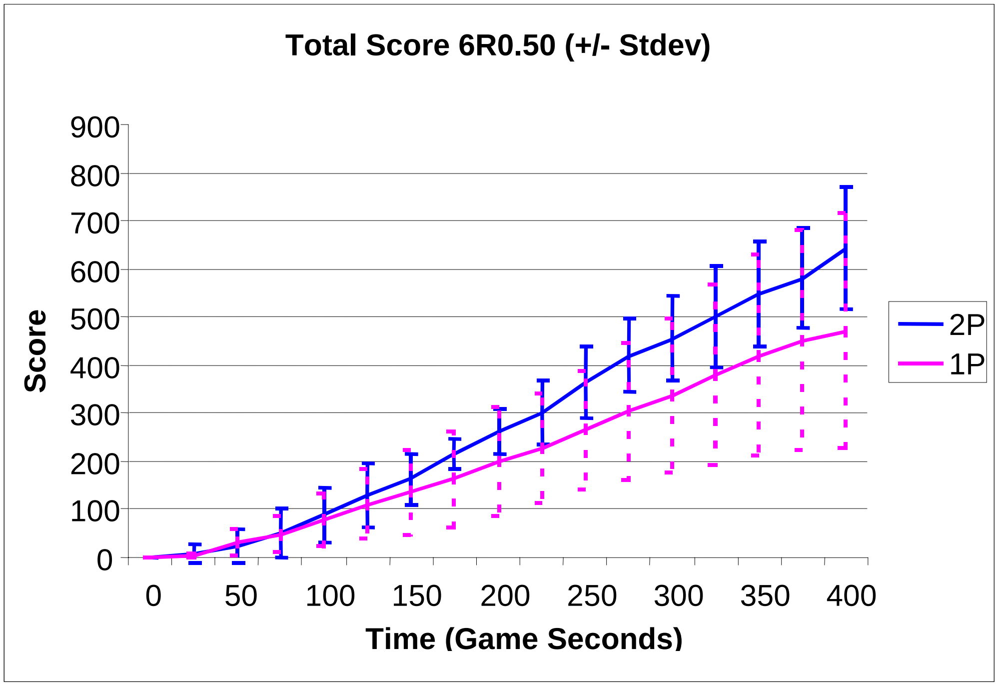


Figure 23: Total score vs. time for one and two players per team (6 robots, 0.50 m/s).

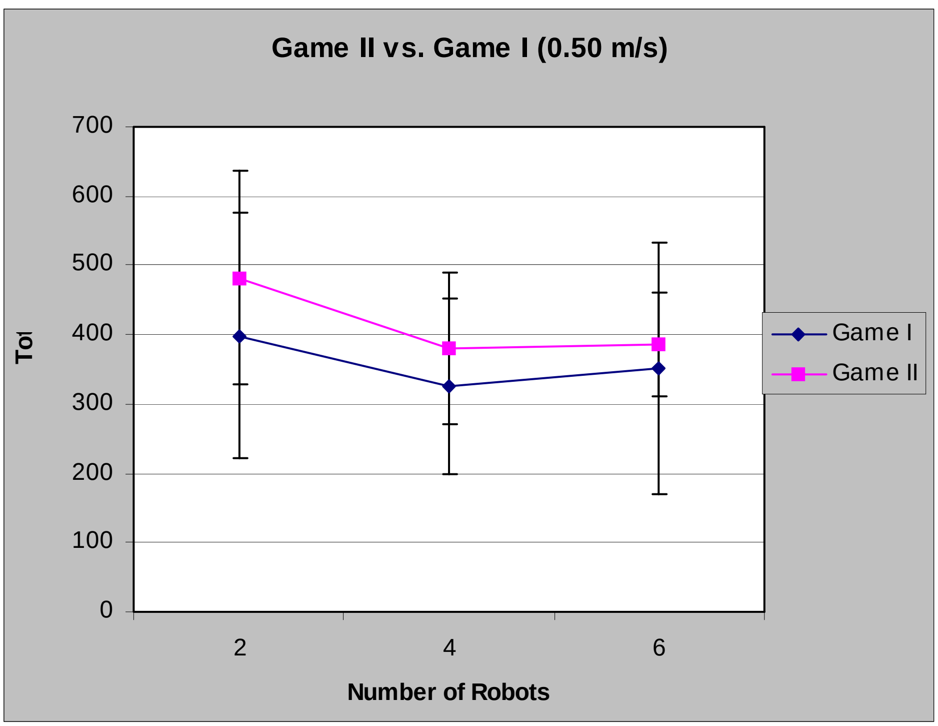


Figure 24: Total score in Phase II Vs. Phase I as a function of number of robots.

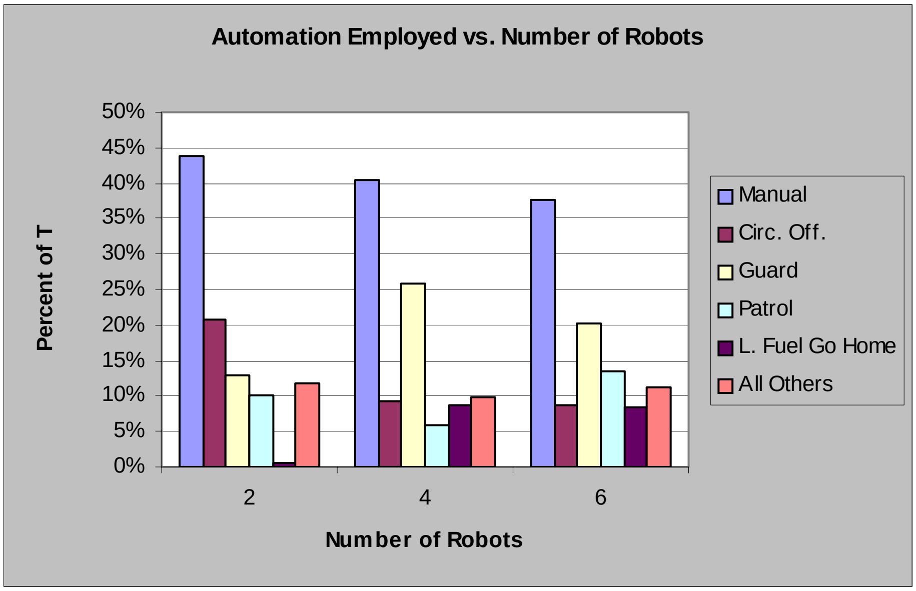


Figure 25: Automation employed versus number of robots at 0.50 m/s for one on one operator games.

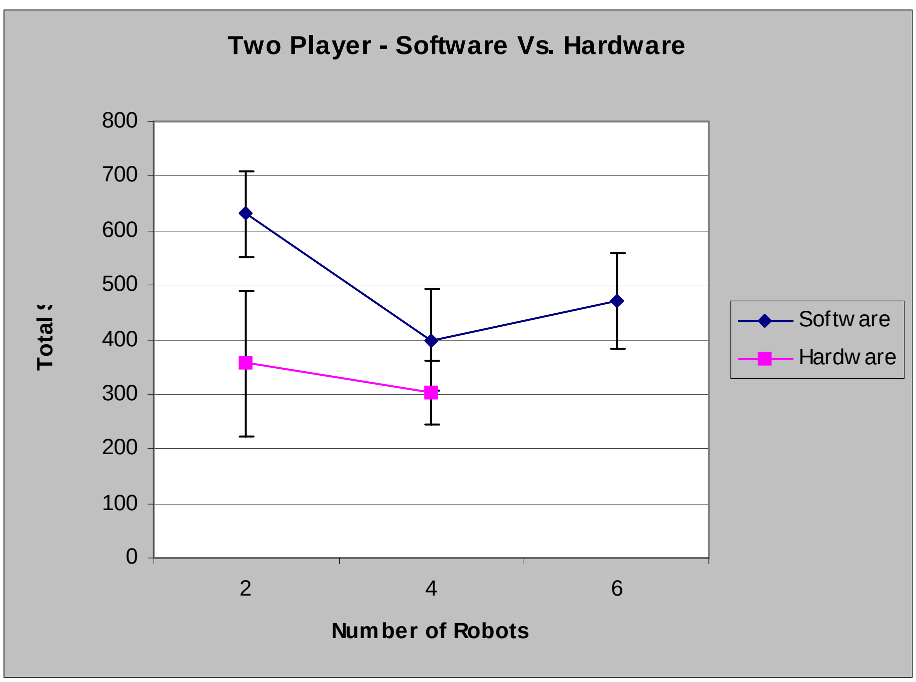


Figure 26: Total Score for two vs. two players as a function of software and hardware implementation.

This study sheds light on the answers to the questions posed previously in the introduction. Based on the results from Phase I and II the following summary has been made:

- Operators “micro-managed” robots more heavily when controlling a smaller number; they relied more on automation as number of robots increased.
- Increasing number of robots was not correlated with a higher score but rather a focus on defensive strategies.
- Increased GUI clicks were correlated with an increase in robots, but only to a modest extent.
- Increased game speed was correlated with an increase in score despite operators complaining of more distrust in the automations and a tendency to use relatively more manual control when possible.
- Cognitive workload remained relatively level as game speed increased, but rose slightly as number of robots increased.
- When automation was improved (Phase II) operators were able to score higher holding other conditions constant.
- Adding an additional human player always increased total score and allowed for less reliance on automation.
- Hardware implementation generated similar results after consideration of poorer performance due to technical problems.

Accordingly the following conclusions can be drawn regarding the design of similar systems of semi-autonomous vehicles and their application to the DARPA MICA challenge problem.

- As the number of vehicles controlled by an operator increases offloading tasks to automation becomes highly important. However, poor automation design can and will lead to little or no increase in operator performance. Additionally defensive strategies become more important suggesting that automation of this aspect is a priority.
- Because GUI input and subjective cognitive workload increase only slightly as number of robots increase, it appears users try to maximize their participation with the system at all times. This suggests issues relating to insufficient situation awareness due to lack of stimulus are minor.
- Improved automation did result in higher end performance underscoring the importance of this aspect of system design.
- Adding operators always improved team performance thus suggesting a method for increasing general system performance independent of automation.
- Results from hardware implementation serve as a reminder of the importance of testing on physical systems in order to capture all aspects of system integration.

It is noted also that the Catholic/AFRL team also ran a set of four tests with AFRL users. A summary of the final report for this subcontract can be found in the appendices.

4.1.5: Architecture for the Evolution of Strategies

Strategic planning assigns vehicles to specific tasks, both *a priori* and during the game. The interface between the strategic planning and path planning, as well as between strategic planning and higher level resource management, can be fuzzy depending on the proposed solution set. In addition, strategic planning could focus on a single small play (cooperative reconnaissance), or a larger methodology (genetic programming based selection of strategies). The approach developed here (to only a small degree because of the abrupt end of the program unfortunately), is to develop a set of lower level plays, and then use these plays to evolve higher level strategies. The evolution process is based on genetic algorithm theory. In summary, the significance of this work is:

Lower level plays used to evolve higher level strategies
- Modular library definition is key to fast computation
- Unique higher level strategies for complex systems
- Based on well known numerical techniques (which are geared towards complex systems)

Useful for:
- Strategy definition
- Resource Management

The concept of team strategizing using lower level plays is given in Figure 27. The hierarchical architecture is given at the left, while a sample of the lower level plays is given at the right. Most of the effort in this area went into defining the lower level plays in the OEP framework. These are given in the appendices.

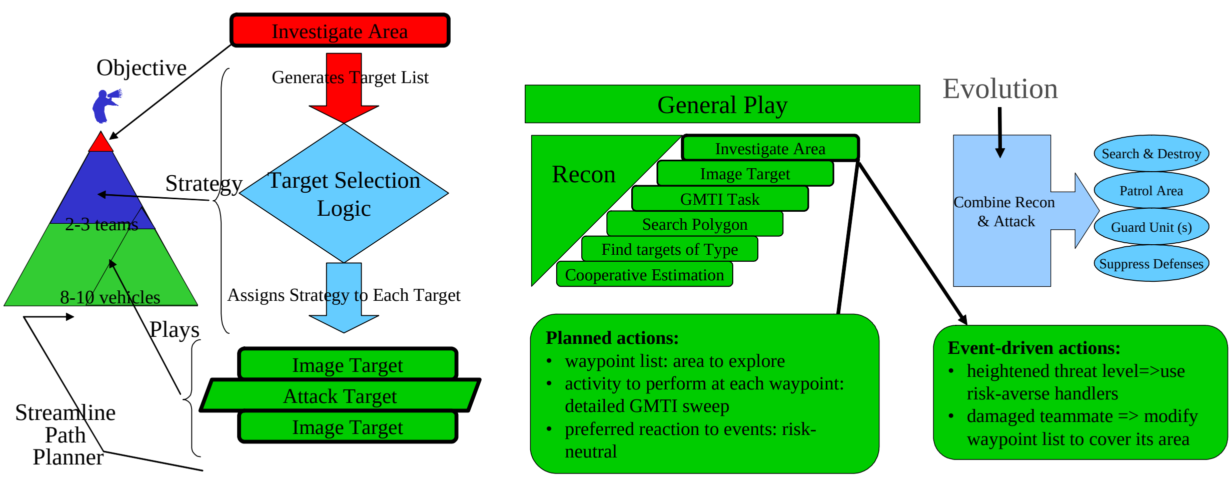


Figure 27: Concept of team strategizing using lower level plays. The hierarchical architecture is given at the left, while a sample of the lower level plays is given at the right.

Figure 28 shows a flow chart of how these plays may evolve. Heuristics are used to define initial strategies or plays. These tactics, along with user defined tasks, are parameterized to allow the

system to be explored with a search based algorithm, such as a genetic program. The results of the search are then used to define strategies for the user, along with a statistical model of its usefulness (such as risk and probability of success). Unfortunately, the genetic evolution process, such as the one shown in Figure 28, was not implemented before the end of the program.

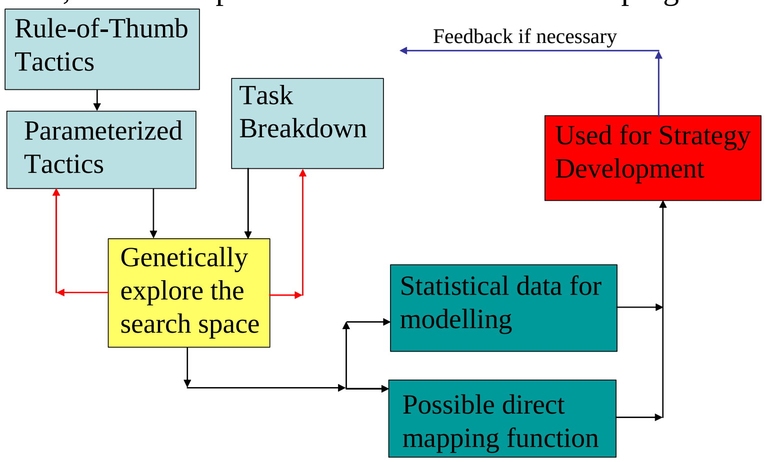


Figure 28: Flow chart describing basic process of creating GP based strategy and resource definition.

4.1.6 Team Tasking using Tiered Optimization

Based on the cooperative reconnaissance work, a team tasking environment has been developed [5]. Plays such as Recon, Defend, Attack can be developed with "team costs", each with timing, signature constraints, etc. Each vehicle can then switch teams if costs shows it to be important, and there are no other constraints violated. This also allows us to quickly explore options of multi-objective (Recon and Strike) using heterogeneous vehicles/functions, such as
2 recon cooperating, 2 attack cooperating, 2 and 1, etc.

The first step was to develop a set of options. This was done using primitives based on the vehicle constraints, as shown in Figure 29.

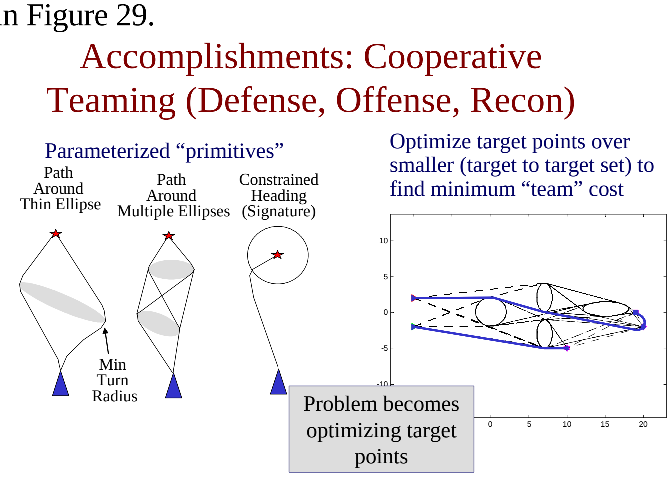


Figure 29: Teaming concepts developed initially by using parameterized vehicle primitives.

The second step was to optimize over the options. Depending on the number of vehicles, three tiers of optimization could be used. These are shown in Figure 30.

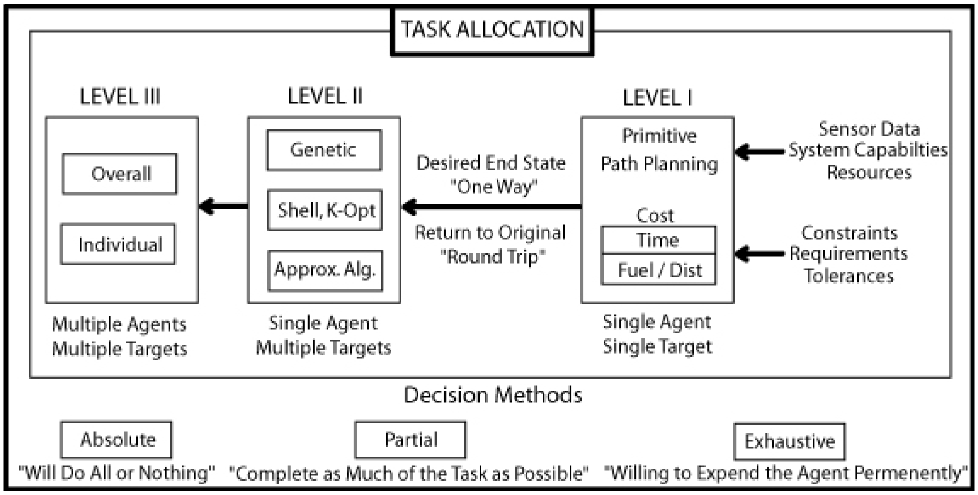


Figure 30: Three tiers of optimization for the team tasking approach based on primitives.

4.2 Research still needed to achieve operational capability

The following is a summary of the of possibilities for future research:

- Theory/studies on how best to enter the human element into the overall system architecture
  - *not addressed fully in MICA*
  - Adaptive tasking based on situation; remain stable
  - Basic research needed for semi-autonomous teams performing complex tasks
  - Research likely be empirical in initial stages; experience will dictate a mathematical framework required to properly frame the key questions.
  - Both modeling and testing
- Packet/Comm based control theory
  - Caltech Multi-Vehicle Wireless Testbed: packet-based communications lose approximately 10% of packets for 6 operating vehicles
  - New control paradigms to handle packetized data
  - Multi-description coding to provide efficient redundancy in presence of packet losses
  - Control signals to control packets
- Spatio-temporal cooperation
  - Move beyond maintaining fixed spatial patterns (formations) and beyond simplified task assignments (rule based)
  - Vehicles maintaining complex spatio-temporal relationships to each other
    - Robust cooperation
    - Example: low probability of detection, situational awareness mission
- Real Time Validation

4.3 Capabilities Achieved

The following capabilities were achieved:

1. A path planner that can run 100 path options in 1/30th of a second, and include aspects such as risk, path length, and integration of path constraints.
2. A cooperative reconnaissance methodology that, based on the OEP, improves Prosecution of moving targets by 20%.

In addition to these specific capabilities, the following additional capabilities have been developed:

3. A hardware and software game of RoboFlag that has allowed the real time validation of technologies for semi-autonomous control. The RoboFlag game has been adopted by over six universities in their research, and was the subject of a very successful invited, interactive session at the 2003 ACC conference.
4. A library of plays to be used for strategy evolution concepts.
5. A generalized teaming approach, still in its development.
6. A series of human in the loop tests, catalogued and summarized in several conference papers, including a set of insights and results.

4.4 Representative Simulation Runs

A typical RoboFlag run is given in the previous section. A OEP run is also developed. Here, the lower level library of plays was used in a typical SEAD type mission. A script of this scenario is given in the appendices. The mission is summarized as follows:

Goal: SEAD Mission (2 Long, 5 Medium)

- Vehicles:
    - Large Sensor platform: ELINT, EO for target location, damage assessment
    - Small Weapon platform: Seeker and anti-radiation missiles to attack sites, self defense; can request confirmation from sensor platform
- Built on streamline path planning about all hostile or unknown sites
- Preplanned:
    - Objectives are known ahead of time
    - A “To-Do” list with constraints on completion time and place; optimized and coordinated between multiple vehicles
- Real time
    - Specific Plans (where, when, how)
    - Dynamic To-Do list is modified with new tasks

The following figures detail a story board of one of the mission sets:

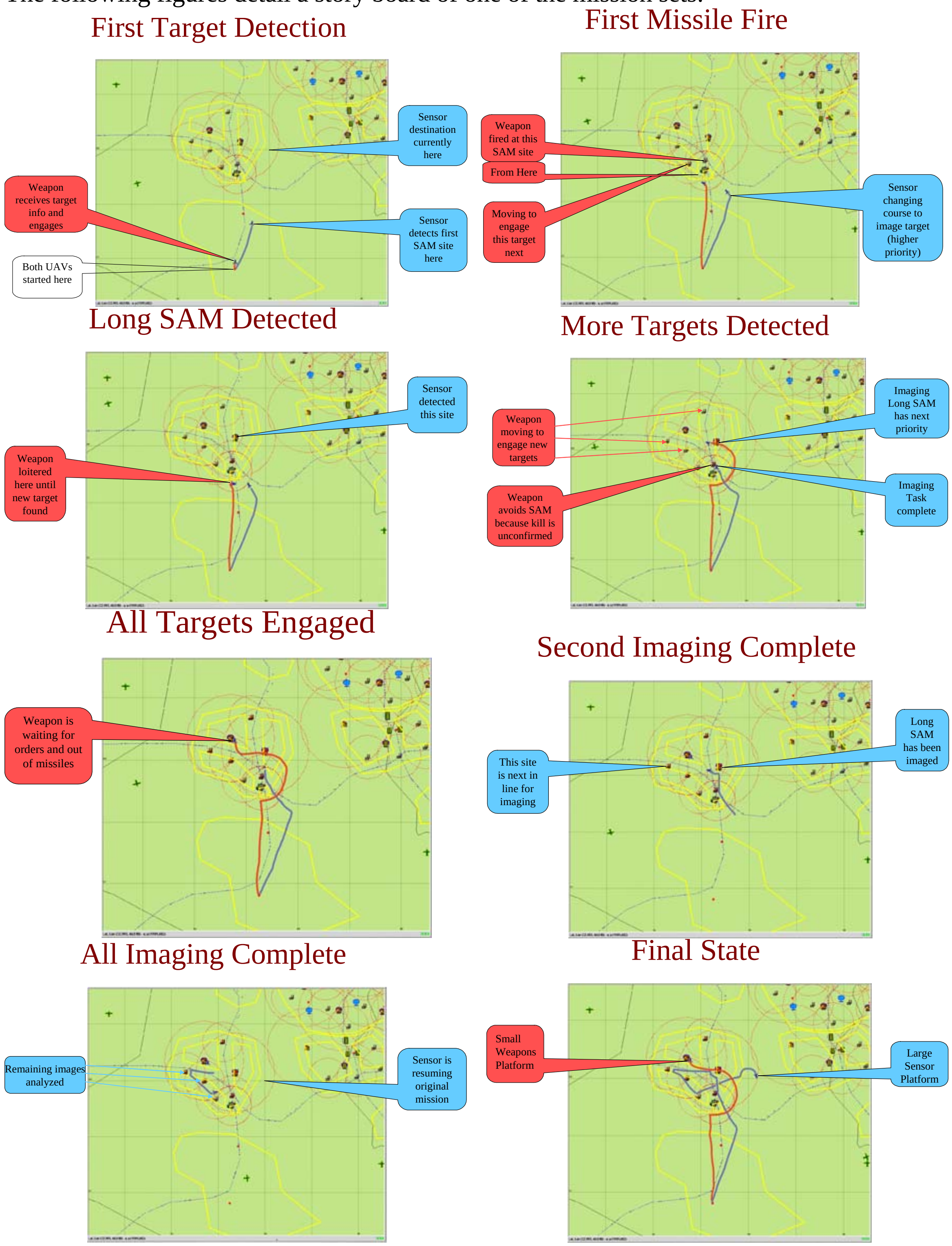


Figure 31: Story board of a typical OEP simulation of the Cornell software.

A summary of the results of running this scenario several times:

Monte Carlo based results (ave of 25 runs)

- Targets Imaged: 8.4
- Targets Destroyed/Damaged: 8.0
- Missiles Fired: 4.7
- UAV's Damaged/Destroyed: 0.36

As another example, the figure below shows how the Cornell software handles a robustness issue: in this case, the weapons effectiveness is overestimated.

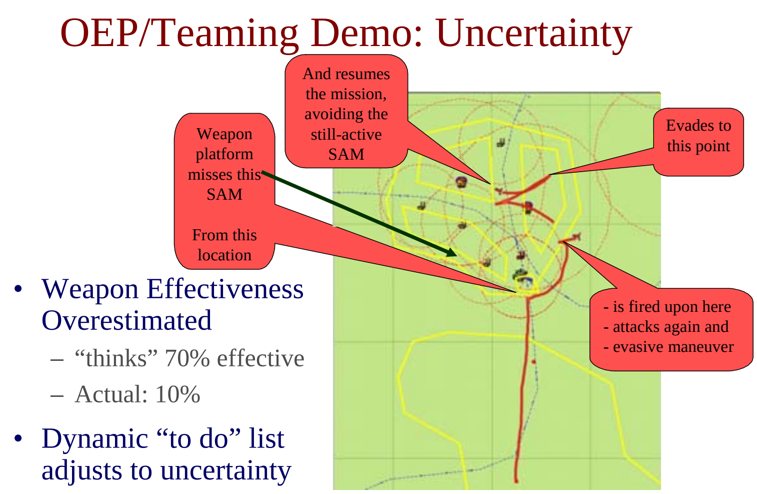


Figure 32: Changing of the weapons effectiveness to understand how the hierarchy handles robustness issues.

CHAPTER 5: CONCLUSIONS

The Cornell led team developed a hierarchical solution to the DARPA MICA problem, and implemented it in real time in hardware. The following capabilities were achieved:

1. A path planner that can run 100 path options in 1/30th of a second, and include aspects such as risk, path length, and integration of path constraints.
2. A cooperative reconnaissance methodology that, based on the OEP, improves Prosecution of moving targets by 20%.

In addition to these specific capabilities, the following additional capabilities have been developed:

3. A hardware and software game of RoboFlag that has allowed the real time validation of technologies for semi-autonomous control. The RoboFlag game has been adopted by over six universities in their research, and was the subject of a very successful invited, interactive session at the 2003 ACC conference.
4. A library of plays to be used for strategy evolution concepts.
5. A generalized teaming approach, still in its development.
6. A series of human in the loop tests, catalogued and summarized in several conference papers, including a set of insights and results.

The following summarizes, in the opinions of the Cornell led, team, key limitations/gaps that remain, what the Cornell team could have accomplished with a full program worth of time, and capabilities that may not be attainable.

Key Limitations/Gaps that Remain:

- Systems level integration and testing
    - Especially real time
- Development with a "truly" intelligent adversary
    - MICA team vs. MICA team would be a start
- Human-centered control of cooperative, multi-vehicle systems.
    - mechanisms by which humans can specify a task to a group of vehicles, modify that task as new information comes in, and understand when changes are required in the high level strategy being used.
    - Adaptive operator tasking
- Political issues: who has the authority?
- Need for more empirical data
    - Robust "playbook" interface evaluation
    - Robust human behavior model
- Full / adaptable autonomy of UV's is not fully achievable at this time given our limited understanding of the circumstances where automation will improve performance while maintaining situation awareness and reducing mental workload

Could/Would have Accomplished:

- Human-centered control of cooperative multi-vehicle systems

    - could have generated ideas for small numbers of operators controlling moderate numbers of vehicles in highly dynamic, adversarial environments.
- HitL experimentation using RoboFlag – assess:
    - Correctness of allocation/requirements decisions
    - Workload across scenario conditions
    - Effect on situation awareness (transitions of control)
    - Tendency to decision bias
    - Trust effects
- Thorough evaluation of the "playbook" interface, which seems to be the best candidate interface for flexible human supervision of multiple robots
- Rigorous evaluation of human performance modeling
    - when combined with empirical data, provides a powerful, objective, science-based rationale for appropriately constraining interface designs

Capabilities Doubt can be Achieved:

- Modeling of operator decisions with enough fidelity to evaluate traditional measure (such as stability)
- "Don't think we got far enough to really know"

CHAPTER 6: RECOMMENDATIONS

The following area areas that the Cornell led team recommends be addressed in future research/work, as they are key technologies that must be developed in order to develop a reliable, semi-autonomous system in practice:

- Theory/studies on how best to enter the human element into the overall system architecture
  - not addressed fully in MICA
  - Adaptive tasking based on situation; remain stable
  - Basic research needed for semi-autonomous teams performing complex tasks
  - Research likely be empirical in initial stages; experience will dictate a mathematical framework required to properly frame the key questions.
  - Both modeling and testing
- Packet/Comm based control theory
  - Caltech Multi-Vehicle Wireless Testbed: packet-based communications lose approximately 10% of packets for 6 operating vehicles
  - New control paradigms to handle packetized data
  - Multi-description coding to provide efficient redundancy in presence of packet losses
  - Control signals to control packets
- Spatio-temporal cooperation
  - Move beyond maintaining fixed spatial patterns (formations) and beyond simplified task assignments (rule based)
  - Vehicles maintaining complex spatio-temporal relationships to each other
  - Robust cooperation
  - Example: low probability of detection, situational awareness mission
- Real time validation of the concepts in a systems level study

REFERENCES

[1] M. Campbell, R. D'Andrea, D. Schneider, A. Chaudhry, et al., "RoboFlag games using systems based hierarchical control" 2003 ACC.

[2] S. Waydo and R. Murray, "Vehicle Motion Planning Using Stream Functions" 2003 ICAR.

[3] J. Sullivan, S. Waydo and M. Campbell, "Using Stream Functions to Generate Complex Behavior" 2003 GNC.

[4] J. Ousingsawat, M. Campbell, "Multiple Vehicle Team tasking for Cooperative Estimation" 2004 ACC (submitted).

[5] Schneider, C. Del Solar, B. Cendek, M. Campbell, "Modular Optimized Task Allocation based on Primitives Path Planning" 2004 ACC (submitted).

[6] L. Chamberlain, J. Tang, M. Watugala, J. Adams and M. Babish, "A Behavioral Architecture for Strategy Execution in the RoboFlag Game" 2003 ACC.

[7] R. D'Andrea and R. Murray, "The RoboFlag Competition" 2003 ACC.

[8] R. D'Andrea and M. Babish, "The RoboFlag Testbed" 2003 ACC.

[9] P. Zigoris, J. Siu, O. Wang and A. Hayes, "Balancing automated behavior and human control in multi-agent systems: a case study in RoboFlag" 2003 ACC.

[10] A. Adams and A. T. Hayes, "The RoboFlag SURF Competition: Results, Analysis, and Future Work" 2003 ACC.

[11] Chaudhry, R. D'Andrea, and M. Campbell, "RoboFlag - A framework for exploring control, planning, and human interface issues related to coordinating multiple robots in a realtime dynamic environment" 2003 ICAR.

[12] J. Veverka and M. Campbell, "Experimental Study of Information Load on Operators in Semi-Autonomous Systems" 2003 GNC.

[13] R. Parasuraman, S. Galster, and C. Miller, "Human Control of Multiple Robots in the RoboFlag Simulation Environment" 2003 IEEE International Conference on Systems, Man, and Cybernetics.

[14] Galster *et al*, 2004 Conference on Human Performance, Situation Awareness and Automation Technology (submitted)

[15] Galster *et al*, HFES 04 (to be submitted)

[16] Z. Jin and R. Murray, "Stability analysis and double-graph model of vehicle formations", 2003 ACC.

[17] L. Cremean, W. Dunbar, D. van Gogh, J. Hickey, E. Klavins, J. Meltzer and R. Murray, "The Caltech Multi-Vehicle Wireless Testbed" 2002 CDC.